DOCUMENT-IDENTIFIER: US 20030105828 A1

TITLE: Systems using Mix of packet, coherent, and noncoherent traffic to optimize transmission between systems

---------- KWIC ---------

Detail Description Paragraph - DETX (9):
[0035] Additionally, in some embodiments, the switch 18 may merge inputs to a given destination virtual channel on a packet boundary. That is, if two sources are requesting to transfer packet data to the same destination and virtual channel, and one of the sources has been granted to that destination and virtual channel, the switch inhibits granting to the other source for that destination and virtual channel until the current source reaches a packet boundary. A similar boundary condition may be used for coherency commands, if more than one transfer through the switch 18 is used to transfer coherency commands. Each of the interfaces 30A-30C used for coherent communications are defined to be capable of transmitting and receiving coherency commands.

Particularly, in the embodiment of FIG. 1, those interfaces 30A-30C may be defined to receive/transmit coherency commands to and from the system 10 from other nodes. Additionally, other types of commands may be carried. In one embodiment, each interface 30A-30C that is used to carry coherency commands may be a HyperTransport.TM. (HT) interface, including an extension to the HT interface to include coherency commands (HTcc). Additionally, in some embodiments, an extension to the HyperTransport interface to carry packet data (Packet over HyperTransport, or PoHT) may be supported. As used herein, coherency commands include any communications between nodes that are used to maintain coherency between nodes. The commands may include read or write requests initiated by a node to fetch or update a cache block belonging to another node, probes to invalidate cached copies of cache blocks in remote nodes (and possibly to return a modified copy of the cache block to the

home node), responses to probe commands, fills which transfer data, etc. A noncoherent command is a communication between devices that does not necessarily occur coherently. For example, standard HT commands may be noncoherent commands.

Detail Description Paragraph - DETX (14):

[0040] In one embodiment, a node such as system 10 may have memory coupled thereto (e.g. memory 24). The node may be responsible for tracking the state, in other nodes, of each cache block from the memory in that node. A node is referred to as the "**home node**" for cache blocks from the memory assigned to that node. A node is referred to as a "remote node" for a cache block if the node is not the **home node** for that cache block. Similarly, a cache block is referred to as a local cache block in the **home node** for that cache block and as a remote cache block in other nodes.

Detail Description Paragraph - DETX (15):

[0041] Generally, a remote node may begin the coherency process by requesting a copy of a cache block from the **home node** of that cache block using a coherency command. The memory bridge 32 in the remote node, for example, may detect a transaction on the interconnect 22 that accesses the cache block and may detect that the remote node does not have sufficient ownership of the cache block to complete the transaction (e.g. it may not have a copy of the cache block at all, or may have a shared copy and may require exclusive ownership to complete the transaction). The memory bridge 32 in the remote node may generate and transmit the coherency command to the **home node** to obtain the copy or to obtain sufficient ownership. The memory bridge 32 in the **home node** may determine if any state changes in other nodes are to be performed to grant the requested ownership to the remote node, and may transmit coherency commands (e.g. probe commands) to effect the state changes. The memory bridge 32 in each node receiving the probe commands may effect the state changes and respond to the probe commands. Once the responses have been received, the memory bridge 32 in the **home node** may respond to the remote node (e.g. with a fill

command including the cache block).

Detail Description Paragraph - DETX (16):
[0042] The remote line directory 34 may be used in the **home node** to track the state of the local cache blocks in the remote nodes. The remote line directory 34 is updated each time a cache block is transmitted to a remote node, the remote node returns the cache block to the **home node**, or the cache block is invalidated via probes. As used herein, the "state" of a cache block in a given node refers to an indication of the ownership that the given node has for the cache block according to the coherency protocol implemented by the nodes. Certain levels of ownership may permit no access, read-only access, or readwrite access to the cache block. For example, in one embodiment, the modified, shared, and invalid states are supported in the internode coherency protocol. In the modified state, the node may read and write the cache block and the node is responsible for returning the block to the **home node** if evicted from the node. In the shared state, the node may read the cache block but not write the cache block without transmitting a coherency command to the **home node** to obtain modified state for the cache block. In the invalid state, the node may not read or write the cache block (i.e. the node does not have a valid copy of the cache block). Other embodiments may use other coherency protocols (e.g. the **MESI** protocol, which includes the modified, shared, and invalid states and an exclusive state in which the cache block has not yet been updated but the node is permitted to read and write the cache block, or the MOESI protocol which includes the modified, exclusive, shared, and invalid states and an owned state which indicates that there may be shared copies of the block but the copy in main memory is stale). In one embodiment, agents within the node may implement the **MESI** protocol for intranode coherency. Thus, the node may be viewed as having a state in the internode coherency and individual agents may have a state in the intranode coherency (consistent with the internode coherency state for the node containing the agent).

Detail Description Paragraph - DETX (18):

[0044] In one embodiment, the remote line directory 34 may be configured to track a subset of the local memory space that may be coherently shared with other nodes. That is, the remote line directory 34 may be configured to track up to a maximum number of cache blocks, where the maximum number is less than the total number of cache blocks that may be coherently shared. In another embodiment, the maximum number may be less than the total number of remote cache entries. The remote line directory may have any structure (e.g. cache-like structures such as direct-mapped, fully associative, set associative, etc.). In one embodiment, the remote line directory 34 may be 16 k entries arranged in an 8 way set associative structure. If a cache block is being accessed by a remote node, and the remote line directory 34 in the **home node** detects a miss for the cache block, an entry is allocated to track the cache block. If the allocated entry is currently allocated to track a second cache block, the memory bridge 32 in the **home node** may generate probes to evict the second cache block from the other nodes (and possibly write back modified data to the **home node,** if applicable).

Detail Description Paragraph - DETX (45):

[0071] The memory bridge 32 in the system 10B generates write transactions (e.g. WrInv) on the interconnect 22 in the system 10B in response to the coherency commands. Since the address A is local to the system 10B, the memory controller 14 in the system 10B may receive the write transactions and write the data transmitted with the write transaction (the packet P1 data) to the memory 24B. It is noted that, if other nodes have copies of the cache blocks being written by the write transactions (as indicated by the remote line directory 34 in the system 10B), the memory bridge 32 in the system 10B may also generate probes to those nodes to invalidate those copies. That is, the WrInv transaction may be a coherent transaction that invalidates cached copies of the cache block updated by the WrInv transaction. The memory bridge

32 may
generate a WrInv transaction responsive to the write command and further
responsive to detecting that the write command is in the **home node** and updates
the entire cache block. Thus, the write commands enter the coherent domain
(i.e., they become coherent) in the **home node** (the system 10B in this example).

Detail Description Paragraph - DETX (106):
[0132] Generally, once a descriptor becomes available for a given input
queue, the packet DMA circuit 16 may request data from the switch (as a destination) for that input queue. Packet data received from the switch for
the input queue is stored in the memory buffer indicated by the descriptor. A
packet may be stored in one or more memory buffers. Once the memory **buffer is**
full or the packet is complete, the packet DMA circuit 16 may update the
descriptor to indicate availability of the packet and may return the descriptor
to software.

Detail Description Paragraph - DETX (127):
[0153] The Flush and Kill commands are probe commands for this embodiment.
The memory bridge 32 at the **home node** of a cache block may issue probe commands
in response to a cRdShd or cRdExc command. The memory bridge 32 at the **home**
node of the cache block may also issue a probe command in response to a transaction for a local cache block, if one or more remote nodes has a copy of
the cache block. The Flush command is used to request that a remote modified
owner of a cache block return the cache block to the **home node** (and invalidate
the cache block in the remote modified owner). The Kill command is used to
request that a remote owner invalidate the cache block. In other embodiments,
additional probe commands may be supported for other state change requests
(e.g. allowing remote owners to retain a shared copy of the cache block).

Detail Description Paragraph - DETX (129):
[0155] The Fill command is the command to transfer data to a remote node
that has transmitted a read command (cRdExc or cRdShd) to the **home**

node. The
Fill command is issued by the memory bridge 32 in the **home node** after the
probes (if any) for a cache block have completed.

Detail Description Paragraph - DETX (132):
[0158] The address space between 40.sub.--0000.sub.--0000 and EF_FFFF_FFFF
is the remote coherent space 148. That is, the address space between
40.sub.--0000.sub.--0000 and EF_FFFF_FFFF is maintained coherent between the
nodes. Each node is assigned a portion of the remote coherent space, and that
node is the **home node** for the portion. As shown in FIG. 1, each node is
programmable with a node number. The node number is equal to the most
significant nibble (4 bits) of the addresses for which that node is the **home
node**, in this embodiment. Thus, the node numbers may range from 4 to E in the
embodiment shown. Other embodiments may support more or fewer node numbers, as
desired. In the illustrated embodiment, each node is assigned a 64 Gigabyte
(GB) portion of the memory space for which it is the **home node**. The size of
the portion assigned to each node may be varied-in other embodiments (e.g.
based on the address size or other factors).

Detail Description Paragraph - DETX (133):
[0159] For a given coherent node, there is an aliasing between the remote
coherent space for which that node is the **home node** and the local address space
of that node. That is, corresponding addresses in the local address space and
the portion of the remote coherent space for which the node is the **home node**
access the same memory locations in the memory 24 of the node (or are memory
mapped to the same I/O devices or interfaces, etc.). For example, the node
having node number 5 aliases the address space 50.sub.--0000.sub.--0000 through
5F_FFFF_FFFF to 00.sub.--0000.sub.--0000 through 0F_FFFF_FFFF respectively
(arrow 146). Internode coherent accesses to the memory 24 at the system 10 use
the node-numbered address space (e.g. 50.sub.--0000.sub.--0000 to 5F_FFFF_FFFF,
if the node number programmed into system 10 is 5) to access cache blocks in
the memory 24. That is, agents in other nodes and agents within the

node that
are coherently accessing cache blocks in the memory use the remote coherent
space, while access in the local address space are not maintained coherent with
other nodes (even though the same cache block may be accessed). Thus the
addresses are aliased, but not maintained coherent, in this embodiment. In
other embodiments, the addresses in the remote coherent space and the
corresponding addresses in the local address space may be maintained coherent.

Detail Description Paragraph - DETX (134):

[0160] A cache block is referred to as local in a node if the cache block is
part of the memory assigned to the node (as mentioned above). Thus, the cache
block may be local if it is accessed from the local address space or the remote
coherent space, as long as the address is in the range for which the node is
the **home node**. Similarly, a transaction on the interconnect 22 that accesses a
local cache block may be referred to as a local transaction or local access. A
transaction on the interconnect 22 that accesses a remote cache block (via the
remote coherent address space outside of the portion for which the node is the
home node) may be referred to as a remote transaction or a remote access.

Detail Description Paragraph - DETX (143):

[0169] If the transaction is remote and uncacheable, then the memory bridge
32 may generate a noncoherent read command on the interfaces 30 to read the
data. For example, a standard HT read command may be used (reference numeral
160). If the remote transaction is cacheable and the response on the
interconnect 22 is exclusive, then the exclusive owner supplies the data for
the read (reference numeral 162). If the remote transaction is cacheable, the
response is not exclusive, the cache block is an L2 cache hit, and the
transaction is either RdShd or the transaction is RdExc and the L2 cache has
the block in the modified state, then the L2 cache 36 supplies the data for the
read (reference numeral 164). Otherwise, the memory bridge 32 initiates a
corresponding read command to the **home node** of the cache block (reference

numeral 166).

Detail Description Paragraph - DETX (147):
[0173] If the transaction is a remote transaction, the transaction is a WrFlush transaction, and the response to the transaction is exclusive, the exclusive owner supplies the data (reference numeral 176). If the remote WrFlush transaction results in a non-exclusive response (shared or invalid), the L2 cache 36 supplies the data of the WrFlush transaction (reference numeral 178). In one embodiment, the L2 cache 36 retains the state of the node as recorded in the **home node**, and the L2 cache 36 uses the WrFlush transaction to evict a remote cache block which is in the modified state in the node. Thus, if another agent has the cache block in the exclusive state, that agent may have a more recent copy of the cache block that should be returned to the **home node**. Otherwise, the L2 cache 36 supplies the block to be returned to the **home node**. In either case, the memory bridge 32 may capture the WrFlush transaction and data, and may perform a WB command to return the cache block to the **home node**.

Detail Description Paragraph - DETX (148):
[0174] If the remote transaction is not a WrFlush transaction, and is not cache coherent, the memory bridge 32 receives the write transaction and performs a non coherent write command (e.g. a standard HT write command) to transmit the cache block to the **home node** (reference numeral 180). If the remote transaction is not a WrFlush transaction, is cache coherent, and is an L2 hit, the L2 cache 36 may update with the data (reference numeral 182).

Detail Description Paragraph - DETX (154):
[0180] The above commands are received by the memory bridge 32 for cache blocks for which the system 10 including the memory bridge 32 is the **home node**. The memory bridge 32 may also receive Flush commands or Kill commands for cache blocks for which the system 10 is a remote node. In response to a Flush

command to the cache block A (reference numeral 260), the memory bridge 32 may initiate a RdKill or RdInv transaction on the interconnect 22. If the local state of the cache block is modified, the memory bridge 32 may transmit a WB command to the **home node,** with the cache block supplied on the interconnect 22 in response to the RdKill or RdInv transaction (reference numeral 262). If the local state of the cache block is not modified, the memory bridge 32 may not respond to the Flush command (reference numeral 264). In this case, the node may already have transmitted a WB command to the **home node** (e.g. in response to evicting the cache block locally). In response to a Kill command to the cache block A (reference numeral 270), the memory bridge 32 may initiate a RdKill or RdInv transaction on the interconnect 22. The memory bridge 32 may respond to the Kill command with a Kill_Ack command (reference numeral 272).

Detail Description Paragraph - DETX (155):

[0181] In one embodiment, the memory bridge 32 may also be configured to receive a non-cacheable read (RdNC) command (e.g. corresponding to a standard HT read) (reference numeral 280). In response, the memory bridge 32 may initiate a RdShd transaction on the interconnect 22. If the RLD state is modified for the cache block including the data to be read, the memory bridge 32 may transmit a Flush command to the remote node having the modified cache block (reference numeral 282), and may receive the WB command from the remote node (reference numeral 284). Additionally, the memory bridge 32 may supply data received on the interconnect 22 in response to the RdShd transaction as a read response (RSP) to the **requesting node** (reference numeral 286).